

LAB 6

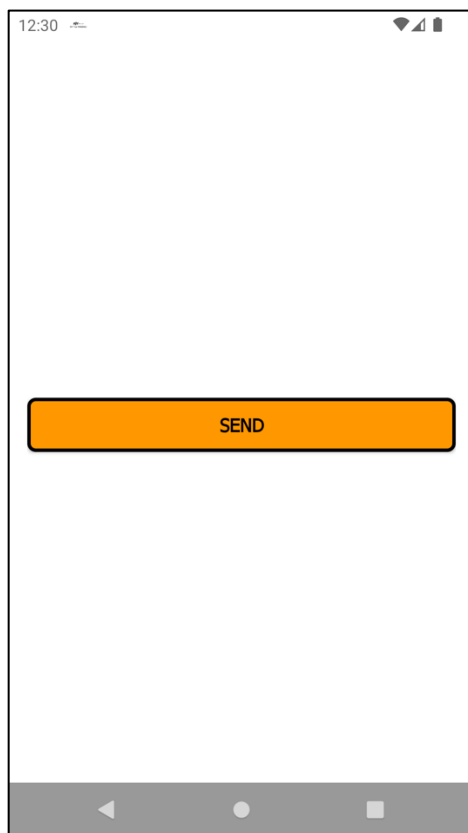
MỤC TIÊU

Kết thúc bài thực hành sinh viên có khả năng:

- ✓ Có thể tạo notification
- ✓ Hiểu rõ cách hoạt động của các loại Services và Worker Manager.

NỘI DUNG

BÀI 1: Tạo một ứng dụng như hình bên dưới, khi nhấn nút Send ứng dụng sẽ tạo ra một notification với nội dung cho trước.



Bước 1: Thực tạo một file config cho Notification

```
public class ConfigNotification extends Application {
    2 usages
    public static final String CHANNEL_ID = "FPTPOLYTECHNIC";

    @Override
    public void onCreate() {
        super.onCreate();
        config();
    }

    1 usage
    private void config(){
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            //Tên của Notification Channel cần đăng ký
            CharSequence name = getString(R.string.channel_name);
            //Mô tả của Notification Channel
            String description = getString(R.string.channel_description);
            //Độ ưu tiên của Notification
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            //Sử dụng RingtoneManager để lấy uri của âm thanh notification theo máy
            Uri uri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_RINGTONE);
            //Tạo thêm một audioAttributes
            AudioAttributes audioAttributes = new AudioAttributes.Builder()
                .setUsage(AudioAttributes.USAGE_NOTIFICATION)
                .build();
            //Đăng ký NotificationChannel
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
            channel.setDescription(description);
            //Set âm thanh cho notification
            channel.setSound(uri, audioAttributes);
            //Đăng ký channel với hệ thống
            NotificationManager notificationManager = getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
}
```

Bước 2: Trong Android Manifest:

- Thêm quyền Post Notification
- Set thuộc tính **name** trong thẻ **<application>**. Giá trị của thuộc tính name chính là tên của file config tạo ở bước 1

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />

    <application
        android:name=".ConfigNotification"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Android02"
```

Bước 3: Tạo một notification mới

```
//Ta sẽ dùng bitmap để đưa hình vào notification
Bitmap logo = BitmapFactory.decodeResource(getResources(), R.mipmap.logofpt);

NotificationCompat.Builder builder = new NotificationCompat.Builder(context: this, ConfigNotification.CHANNEL_ID)
    //icon sẽ hiện thị trên status bar
    .setSmallIcon(R.mipmap.logofpt)
    //Tiêu đề của notification
    .setContentTitle("Chào mừng đến với FPT Polytechnic")
    //Nội dung của notification
    .setContentText("Android 2")
    //Truyền một hình ảnh vào notification
    .setStyle(new NotificationCompat.BigPictureStyle()
        .bigPicture(logo)
        .bigLargeIcon(b: null)
    )
    //Hiện thị icon bên phải khi notification ở dạng thu gọn
    .setLargeIcon(logo)
    .setColor(Color.RED)
    .setAutoCancel(true);

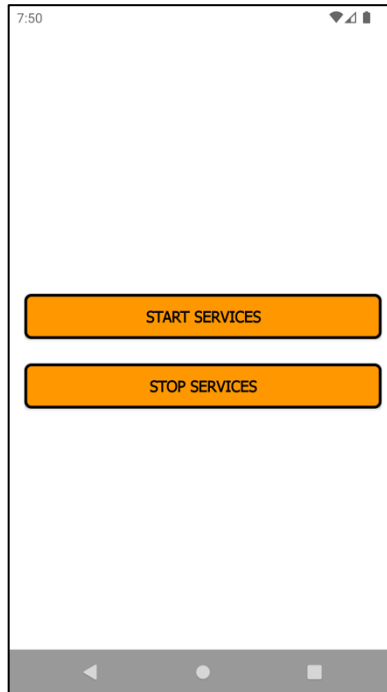
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(context: this);
//Code kiểm tra quyền notification trên thiết bị
if (ActivityCompat.checkSelfPermission(context: this, Manifest.permission.POST_NOTIFICATIONS)
    == PackageManager.PERMISSION_GRANTED) {
    //Nếu đã có quyền, ta thực hiện push notification
    //mỗi thông báo được push cần có 1 id riêng, sử dụng Date().getTime() để tạo ID cho thông báo
    notificationManager.notify((int) new Date().getTime(), builder.build());
} else {
    //Nếu ko có quyền, thì sẽ thực hiện xin quyền
    ActivityCompat.requestPermissions(activity: this,
        new String[]{Manifest.permission.POST_NOTIFICATIONS}, requestCode: 7979);
}
```

Bước 4: Tạo hàm xin quyền nếu ứng dụng chưa được cấp quyền thông báo

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 7979) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
                sendNotification();
            }
        }
    }
}
```

BÀI 2: Tạo và dừng Foreground Service

Bước 1: Tạo một màn hình có layout gồm 2 nút start Service, stop Service



Bước 2: Tạo một file mới đặt tên là **ForegroundService**, trong file này ta **extends Service** và **implement onBind()**

```
public class ForegroundService extends Service {  
    @Nullable  
    @Override  
    public IBinder onBind(Intent intent) { return null; }  
}
```

Ta tiếp tục override hàm **onStartCommand**, trong hàm này tạo một notification cho Foreground Service.

Ta sử dụng **startForeground** để chạy Service

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    //Bạn có thể sử dụng intent (hoặc Bundle) để nhận dữ liệu trong Service
    Toast.makeText(context: this, text: "Services đang chạy", Toast.LENGTH_SHORT).show();

    NotificationCompat.Builder builder = new NotificationCompat.Builder(context: this, MyApplication.CHANNEL_ID)
        .setSmallIcon(R.mipmap.logofpt)
        .setContentTitle("Services Đang chạy")
        .setContentText("Bạn không thể tắt notification bằng cách lướt nó")
        .setColor(Color.RED);
    //Ta sẽ tạo ra một Notification và truyền nó vào startForeground services
    Notification notification = (Notification) builder.build();
    //Id của startForeground sẽ là số lớn hơn 0
    startForeground(id: 1, notification);

    return START_NOT_STICKY;
}
```

Bước 3: Tiếp tục override hàm **onDestroy** để toast thông báo khi services stop

```
@Override
public void onDestroy() {
    super.onDestroy();
    Toast.makeText(context: this, text: "Destroy Services", Toast.LENGTH_SHORT).show();
}
```

Bước 4: Trong **Android Manifest** thêm một permission **FOREGROUND_SERVICE**

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />

    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />

</manifest>
```

Và khai báo **ForegroundService** tạo ở bước 2 vào trong Application

```
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

    <service android:name=".ForegroundService" />

</application>
```

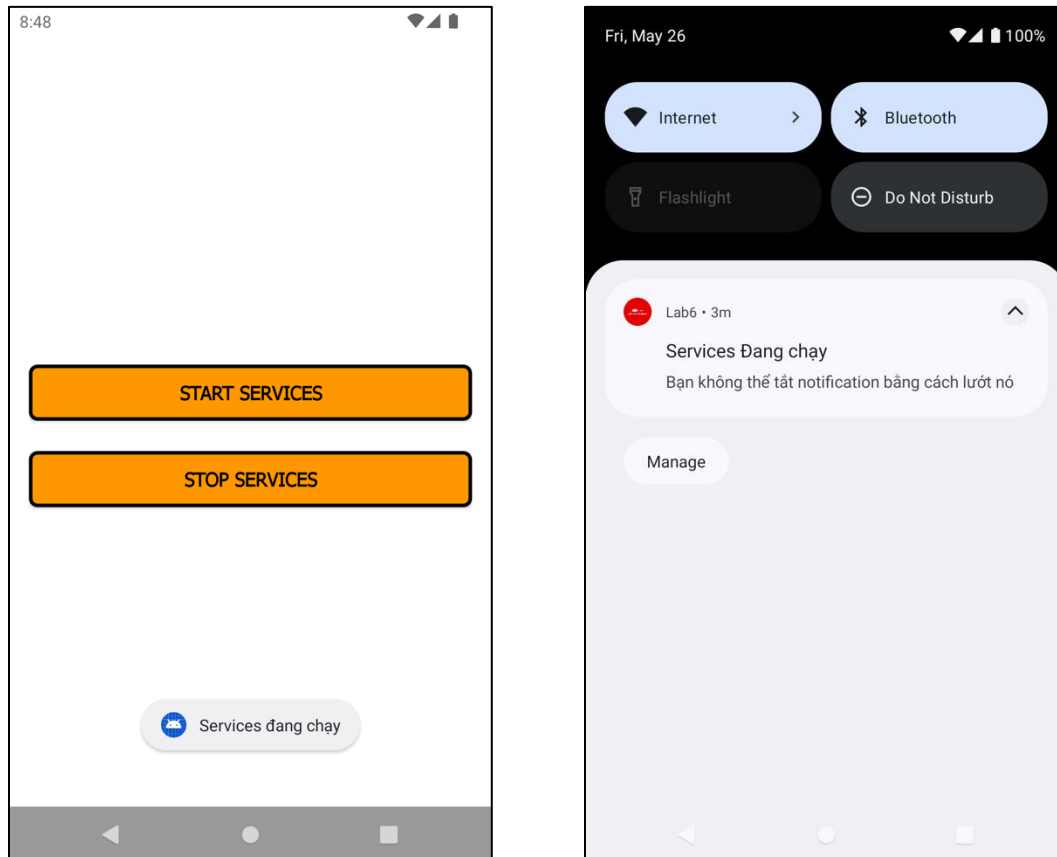
Bước 5: Sử dụng **startService** để chạy Service

```
Intent intent = new Intent( packageContext: MainActivity.this, ForegroundService.class);
//Bạn có sử dụng intent (Hoặc Bundle) để truyền dữ liệu xuống service
startService(intent);
```

Sử dụng **stopService** để dừng Service

```
Intent intent = new Intent( packageContext: MainActivity.this, ForegroundService.class);
stopService(intent);
```

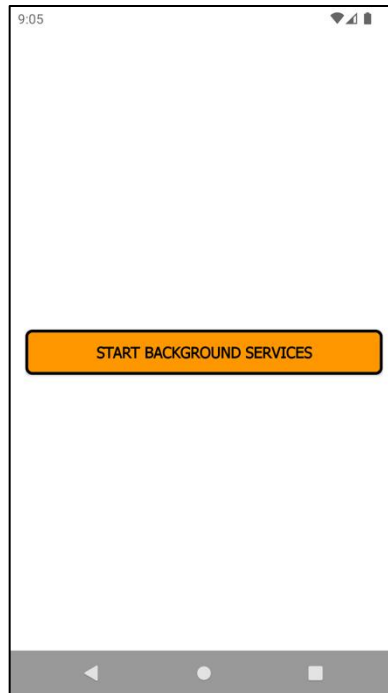
Ta được kết quả như hình bên dưới khi chạy ForegroundService, **notification được tạo ra sẽ không thể tắt hoặc xóa đi khi start Service:**



Khi ta nhấn nút stop Service, notification sẽ tự động tắt

BÀI 3: Tạo một Background Services, khi chạy service sau 5 giây sẽ tự động chuyển sang browser và mở một trang web bất kì

Bước 1: Tạo một màn hình có layout như bên dưới



Bước 2: Tạo một file mới đặt tên là **BackgroundServices**, trong file này ta **extends Service** và **implement onBind()**

```
public class BackgroundServices extends Service {  
    @Nullable  
    @Override  
    public IBinder onBind(Intent intent) { return null; }  
}
```

Ta tiếp tục override hàm **onStartCommand**, trong hàm này ta xử lý chức năng mở một website bất kỳ sau 5 giây

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Toast.makeText(context: this, text: "Service bắt đầu chạy", Toast.LENGTH_SHORT).show();
    try {
        //Sau 5 giây sẽ bắt đầu chuyển sang website muốn đến
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Toast.makeText(context: BackgroundServices.this, text: "Đang chuyển", Toast.LENGTH_SHORT).show();
                Intent webIntent = new Intent(Intent.ACTION_VIEW);
                webIntent.setData(Uri.parse(uriString: "https://caodang.fpt.edu.vn/"));
                webIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(webIntent);
                //Sau khi thực hiện xong chúng ta sẽ đóng service lại
                stopSelf();
            }
        }, delayMillis: 5000);
    } catch (Exception e) {
        Toast.makeText(context: this, text: "Lỗi chuyển qua web site", Toast.LENGTH_SHORT).show();
        //Nếu lỗi sẽ stop services
        stopSelf();
    }
    return super.onStartCommand(intent, flags, startId);
}
```

Bước 3: Tiếp tục override hàm **onDestroy** để toast thông báo khi **services stop**

```
@Override
public void onDestroy() {
    super.onDestroy();
    Toast.makeText(context: this, text: "Destroy Services", Toast.LENGTH_SHORT).show();
}
```

Bước 4: Trong **Android Manifest** thêm permission

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />

    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>

    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <application
        android:name=".MyApplication"
```

Và khai báo **BackgroundServices** tạo ở bước 2 vào trong Application

```
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<service android:name=".ForegroundService"/>
<service android:name=".BackgroundServices"/>
</application>
```

Bước 5: Sử dụng **startService** để chạy Service

```
Intent intent = new Intent( packageContext: MainActivity.this, BackgroundServices.class);
startService(intent);
```

BÀI 4: Tạo một WorkManager thực hiện công việc hiển thị thông báo khi thiết bị được kết nối với sạc sau 1 giây

Bước 1: Vào build.gradle (Module :app) thêm thư viện work-runtime

```
implementation "androidx.work:work-runtime:2.8.1"
```



Bước 2: Tạo một file mới đặt tên là MyWorker, ta extends Workers và tạo constructor cho class

```
public class MyWorker extends Worker {  
    public MyWorker(@NonNull Context context, @NonNull WorkerParameters workerParameters)  
    {  
        super(context, workerParameters);  
    }  
}
```

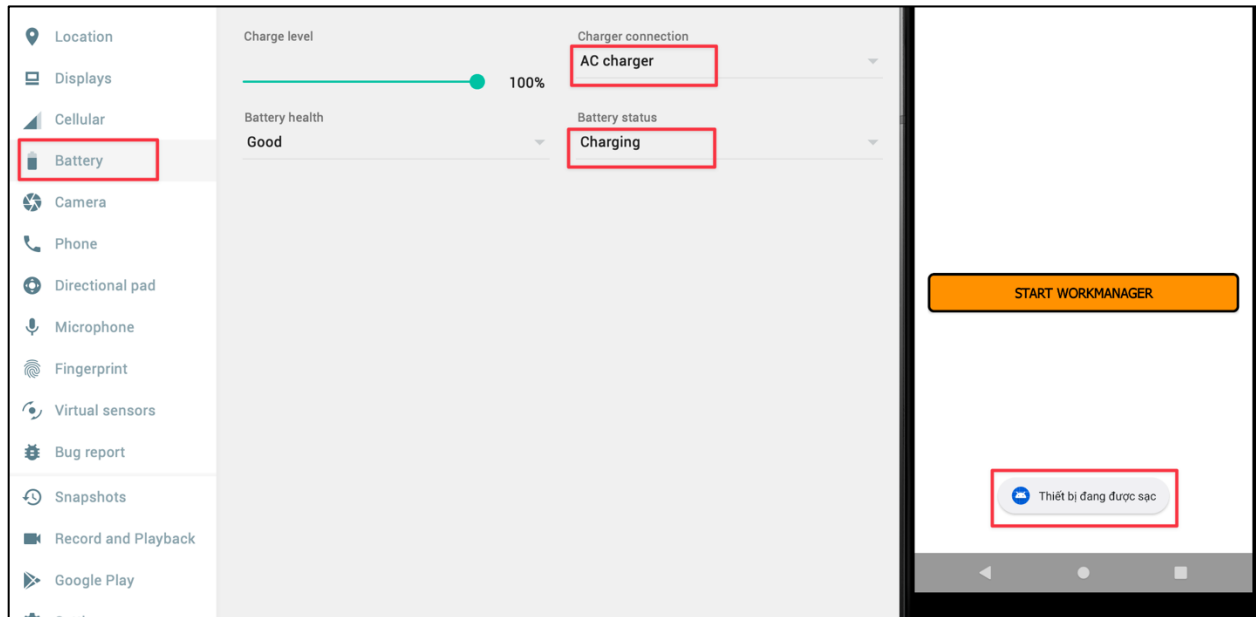
Bước 3: Ta tiếp tục override hàm **doWork()**, trong hàm này ta thực hiện công việc hiển thị thông báo sau 1 giây khi người dùng cắm sạc

```
@NonNull
@Override
public Result doWork() {
    //Tạo handler
    Handler handler = new Handler(Looper.getMainLooper());
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            //Sau 1 giây sẽ Toast lên khi thiết bị được sạc
            Toast.makeText(getApplicationContext(), text: "Thiết bị đang được sạc", Toast.LENGTH_SHORT).show();
        }
    }, delayMillis: 1000);
    return Result.success();
}
```

Bước 4: Đăng ký WorkManager với hệ thống

```
//Khi ứng dụng được kết nối với sạc sẽ thực thi công việc
Constraints constraints = new Constraints.Builder()
    .setRequiresCharging(true).build();
WorkRequest myWorkRequest = new OneTimeWorkRequest.Builder(MyWorker.class)
    .setConstraints(constraints).build();
//Gửi workRequest đến hệ thống
WorkManager.getInstance(context: MainActivity.this).enqueue(myWorkRequest);
```

Bước 5: Chạy thử kết quả



BÀI 5: GV CHO THÊM

*** YÊU CẦU NỘP BÀI:

Sv nén file bao gồm các yêu cầu đã thực hiện trên, nộp lms đúng thời gian quy định của giảng viên. Không nộp bài coi như không có điểm.

--- Hết ---